![1-Logo](Images/1-Logo.png)

Welcome to the United States Geological Survey, or USGS for short! The USGS is responsible for providing scientific data about natural hazards, the health of our ecosystems and environment; and the impacts of climate and land-use change. Their scientists develop new methods and tools to supply timely, relevant, and useful information about the Earth and its processes. As a new hire, you will be helping them out with an exciting new project!

The USGS is interested in building a new set of tools that will allow them visualize their earthquake data. They collect a massive amount of data from all over the world each day, but they lack a meaningful way of displaying it. Their hope is that being able to visualize their data will allow them to better educate the public and other government organizations (and hopefully secure more funding..) on issues facing our planet.

## Your Task

### Level 1: Basic Visualization

![2-BasicMap](Images/2-BasicMap.png)

Your first task is to visualize an earthquake data set.

1. \*\*Get your data set\*\*

![3-Data](Images/3-Data.png)

The USGS provides earthquake data in a number of different formats, updated every 5 minutes. Visit the [USGS GeoJSON Feed](http://earthquake.usgs.gov/earthquakes/feed/v1.0/geojson.php) page and pick a data set to visualize. When you click on a data set, for example 'All Earthquakes from the Past 7 Days', you will be given a JSON representation of that data. You will be using the URL of this JSON to pull in the data for our visualization.

![4-JSON](Images/4-JSON.png)

2. \*\*Import & Visualize the Data\*\*

Create a map using Leaflet that plots all of the earthquakes from your data set based on their longitude and latitude.

\* Your data markers should reflect the magnitude of the earthquake in their size and color. Earthquakes with higher magnitudes should appear larger and darker in color.

\* Include popups that provide additional information about the earthquake when a marker is clicked.

\* Create a legend that will provide context for your map data.

\* Your visualization should look something like the map above.

- - -

### Level 2: More Data (Optional)

![5-Advanced](Images/5-Advanced.png)

The USGS wants you to plot a second data set on your map to illustrate the relationship between tectonic plates and seismic activity. You will need to pull in a second data set and visualize it along side your original set of data. Data on tectonic plates can be found at <https://github.com/fraxen/tectonicplates>.

In this step we are going to..

\* Plot a second data set on our map.

\* Add a number of base maps to choose from as well as separate out our two different data sets into overlays that can be turned on and off independently.

\* Add layer controls to our map.

- - -

### Assessment

Your final product will be assessed on the following metrics:

\* Completion of assigned tasks

\* Visual appearance

\* Professionalism

\*\*Good luck!\*\*

## Copyright

Data Boot Camp (C) 2018. All Rights Reserved.

**{**"type": "FeatureCollection","metadata": **{**"generated": **1563006083000**,"url": "<https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/significant_hour.geojson>","title": "USGS Significant Earthquakes, Past Hour","status": **200**,"api": "1.8.1","count": **0}**,"features": **[]}**

**Or more than 1 hour:**

**{**"type": "FeatureCollection","metadata": **{**"generated": **1563006270000**,"url": "<https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/4.5_hour.geojson>","title": "USGS Magnitude 4.5+ Earthquakes, Past Hour","status": **200**,"api": "1.8.1","count": **1}**,"features": **[{**"type": "Feature","properties": **{**"mag": **5.6**,"place": "Southern East Pacific Rise","time": **1563004775218**,"updated": **1563006134446**,"tz": **-480**,"url": "<https://earthquake.usgs.gov/earthquakes/eventpage/us70004ji2>","detail": "<https://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/us70004ji2.geojson>","felt": **null**,"cdi": **null**,"mmi": **0**,"alert": "green","status": "reviewed","tsunami": **0**,"sig": **482**,"net": "us","code": "70004ji2","ids": ",us70004ji2,","sources": ",us,","types": ",geoserve,losspager,origin,phase-data,shakemap,","nst": **null**,"dmin": **29.939**,"rms": **0.97**,"gap": **102**,"magType": "mww","type": "earthquake","title": "M 5.6 - Southern East Pacific Rise"**}**,"geometry": **{**"type": "Point","coordinates": **[-116.5658**,**-49.597**,**10]}**,"id": "us70004ji2"**}]}**

//Creating a layer ro hold the map

var myMap = L.map("map",{

center: [47, -110],

zoom: 3

});

L.tileLayer("https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png?access\_token={accessToken}", {

attribution: "Map data &copy; <a href=\"https://www.openstreetmap.org/\">OpenStreetMap</a> contributors, <a href=\"https://creativecommons.org/licenses/by-sa/2.0/\">CC-BY-SA</a>, Imagery © <a href=\"https://www.mapbox.com/\">Mapbox</a>",

maxZoom: 18,

id: "mapbox.streets",

accessToken: API\_KEY

}).addTo(myMap);

var usgsUrl = "https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2014-01-01&endtime=2014-01-02";

//Storing the query from USGS

d3.json(usgsUrl, function (detailData) {

//add a marker

var marker = L.markerclusterGroup();

//loop into the data

for (var i=0; i< detailData.length; i++) {

//set the data location property into the data

var location = detailData[i].location;

// reck for the location property

if (location) {

//adding a new marker to the cluster group and bind it with a pop up

marker.addLayer(L.marker([location.coordinates[1], location.coordinates[0]])

.bindPopup(detailData[i].descriptor));

}

}

//detailDesign(detailData.features);

L.geoJson(detailData).addTo(myMap);

myMap.addLayer(marker);

});

//Creating a Json Layer for the features, with the capacity to describe

// the place and time for the earthquake.

//function detailDesign(usgsData) {

// function onEachFeature(features, layer) {

// layer.bindPopup("<h2>" + features, properties.place +

// "</h2><hr><p>" + new Date(feature.properties.tima) + "</p>");

// }

//Creating a Json layer

// var earthquakes = L.geoJSON(usgsData, {

// onEachFeature: onEachFeature

// });

//compiling all layers to create a map function

// createImageBitmap(earthquakes);

//}

L.tileLayer("https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png?access\_token={accessToken}", {

attribution: "Map data &copy; <a href=\"https://www.openstreetmap.org/\">OpenStreetMap</a> contributors, <a href=\"https://creativecommons.org/licenses/by-sa/2.0/\">CC-BY-SA</a>, Imagery © <a href=\"https://www.mapbox.com/\">Mapbox</a>",

maxZoom: 18,

id: "mapbox.streets",

accessToken: API\_KEY

}).addTo(myMap);

//L.control.layers(baseMaps, overlayMaps, {

// collapsed: false

// }).addTo(myMap);

|  |
| --- |
| // Store our API endpoint inside queryUrl |
|  | var queryUrl = "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all\_week.geojson" |
|  |  |
|  | var query2 = "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5\_week.geojson" |
|  |  |
|  | // Perform a GET request to the query URL |
|  | d3.json(queryUrl, function(data) { |
|  | // Once we get a response, send the data.features object to the createFeatures function |
|  | createFeatures(data.features); |
|  | }); |
|  |  |
|  | function createFeatures(earthquakeData) { |
|  |  |
|  |  |
|  | // Give each feature a popup describing the place and time of the earthquake |
|  | function onEachFeature(feature, layer) { |
|  | layer.bindPopup("<h3>" + feature.properties.place + |
|  | "</h3><hr><p>" + new Date(feature.properties.time) + "</p>" + |
|  | "</h3><hr><p>Magnitude: " + feature.properties.mag + "</p>"); |
|  | } |
|  |  |
|  |  |
|  |  |
|  | // Create a GeoJSON layer containing the features array on the earthquakeData object |
|  | // Run the onEachFeature function once for each piece of data in the array |
|  | var earthquakes = L.geoJSON(earthquakeData, { |
|  | onEachFeature: onEachFeature, |
|  | pointToLayer: function (feature, latlng) { |
|  | var color; |
|  | var r = 255; |
|  | var g = Math.floor(255-80\*feature.properties.mag); |
|  | var b = Math.floor(255-80\*feature.properties.mag); |
|  | color= "rgb("+r+" ,"+g+","+ b+")" |
|  |  |
|  | var geojsonMarkerOptions = { |
|  | radius: 4\*feature.properties.mag, |
|  | fillColor: color, |
|  | color: "black", |
|  | weight: 1, |
|  | opacity: 1, |
|  | fillOpacity: 0.8 |
|  | }; |
|  | return L.circleMarker(latlng, geojsonMarkerOptions); |
|  | } |
|  | }); |
|  |  |
|  |  |
|  | // Sending our earthquakes layer to the createMap function |
|  | createMap(earthquakes); |
|  |  |
|  | } |
|  |  |
|  | function createMap(earthquakes) { |
|  |  |
|  | // Define streetmap and darkmap layers |
|  | var streetmap = L.tileLayer("https://api.mapbox.com/styles/v1/mapbox/outdoors-v10/tiles/256/{z}/{x}/{y}?" + |
|  | "access\_token=pk.eyJ1Ijoia2pnMzEwIiwiYSI6ImNpdGRjbWhxdjAwNG0yb3A5b21jOXluZTUifQ." + |
|  | "T6YbdDixkOBWH\_k9GbS8JQ"); |
|  |  |
|  | // Define a baseMaps object to hold our base layers |
|  | var baseMaps = { |
|  | "Street Map": streetmap |
|  | }; |
|  |  |
|  | // Create overlay object to hold our overlay layer |
|  | var overlayMaps = { |
|  | Earthquakes: earthquakes |
|  | }; |
|  |  |
|  | // Create our map, giving it the streetmap and earthquakes layers to display on load |
|  | var myMap = L.map("map", { |
|  | center: [ |
|  | 37.09, -95.71 |
|  | ], |
|  | zoom: 5, |
|  | layers: [streetmap, earthquakes] |
|  | }); |
|  |  |
|  |  |
|  | function getColor(d) { |
|  | return d < 1 ? 'rgb(255,255,255)' : |
|  | d < 2 ? 'rgb(255,225,225)' : |
|  | d < 3 ? 'rgb(255,195,195)' : |
|  | d < 4 ? 'rgb(255,165,165)' : |
|  | d < 5 ? 'rgb(255,135,135)' : |
|  | d < 6 ? 'rgb(255,105,105)' : |
|  | d < 7 ? 'rgb(255,75,75)' : |
|  | d < 8 ? 'rgb(255,45,45)' : |
|  | d < 9 ? 'rgb(255,15,15)' : |
|  | 'rgb(255,0,0)'; |
|  | } |
|  |  |
|  | // Create a legend to display information about our map |
|  | var legend = L.control({position: 'bottomright'}); |
|  |  |
|  | legend.onAdd = function (map) { |
|  |  |
|  | var div = L.DomUtil.create('div', 'info legend'), |
|  | grades = [0, 1, 2, 3, 4, 5, 6, 7, 8], |
|  | labels = []; |
|  |  |
|  | div.innerHTML+='Magnitude<br><hr>' |
|  |  |
|  | // loop through our density intervals and generate a label with a colored square for each interval |
|  | for (var i = 0; i < grades.length; i++) { |
|  | div.innerHTML += |
|  | '<i style="background:' + getColor(grades[i] + 1) + '">&nbsp&nbsp&nbsp&nbsp</i> ' + |
|  | grades[i] + (grades[i + 1] ? '&ndash;' + grades[i + 1] + '<br>' : '+'); |
|  | } |
|  |  |
|  | return div; |
|  | }; |
|  |  |
|  | legend.addTo(myMap); |
|  |  |
|  | } |